

## (57) Abstract

A system and method to eliminate the critical path in a processor-based system by sending a signal to transition the memory and I/O control unit to the read/write state before instruction decode completes up to the end. The decode stage of the operation code of the instructions, if it is made clear that the step of reading and writing in which the memory or I/O device must be accessed should be executed, the processor immediately sends a read/write request to the memory control unit and the I/O control unit before decoding the remaining part of the instruction. Once the remaining part of the instruction has been decoded, if it has been decided that the storage place that is accessed is in either the memory or the I/O device, cancellation processing occurs. In this cancellation processing, if the storage place that is accessed is within the memory, the I/O unit transitions from the read/write state to the idle state. However, if the access destination is the I/O device, the memory control unit transitions from the read/write state to the idle state.

(51) Int. Cl. <sup>4</sup>	識別記号	庁内整理番号	F I
G 0 6 F 9/38	3 5 0 2	9193-5 B	
9/34	3 5 0 A	9189-5 B	
12/00	5 6 0 C	9366-5 B	

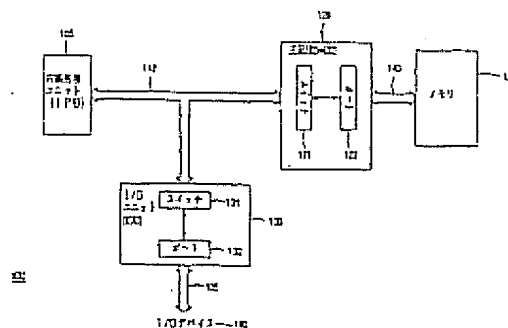
(21) 出願番号	特願平5-515523	(71) 出願人	セイコーエプソン株式会社 東京都新宿区西新宿2丁目4番1号
(26) (22) 出願日	平成5年(1993)3月2日	(72) 発明者	ミヤヤマ ヨシユキ アメリカ合衆国 95050 カリフォルニア 州 サンタ クララ、ランチョ マクコー ミック ブルバード 2171
(85) 翻訳文提出日	平成6年(1994)9月2日	(72) 発明者	タン チェンローン アメリカ合衆国 95131 カリフォルニア 州 サン ホゼ、リビシ ウェイ 1915
(86) 国際出願番号	P C T / J P 9 3 / 0 0 2 5 9	(74) 代理人	弁理士 鈴木 孝三郎 (外1名)
(87) 国際公開番号	W O 9 3 / 1 8 4 5 1		
(87) 国際公開日	平成5年(1993)9月16日		
(31) 優先権主張番号	8 4 6 . 2 3 1		
(32) 優先日	1992年3月6日		
(33) 優先権主張国	米国 (U S)		
(81) 指定国	J P		

審査請求 未請求 予備審査請求 有 (全 10 頁)

(54) 【発明の名称】 メモリ制御ユニット並びに入出力制御ユニットの動作におけるクリティカル・バスの削除

## (57) 【要約】

命令のデコードが最後まで完了してしまう前に、メモリやI/O制御ユニットを読み出し/書き込み状態に移すために信号を送ることによって、プロセッサ・ベースのシステムにおいてクリティカル・バスを削除するためのシステム並びに方法。命令の演算コードのデコード段階が、メモリまたはI/Oデバイスがアクセスされなければならない読み出し書き込みのステップが実行されるべきであることを明らかにすれば、プロセッサは、命令の残りの部分をデコードする前に、直ちに読み出し書き込み要求をメモリ制御ユニット及びI/O制御ユニットに送る。一旦命令の残りの部分がデコードされ、アクセスする記憶場所がメモリまたはI/Oデバイスのどちらかにあることが決定されれば、取消し処理が起きる。この取消し処理において、アクセスする記憶場所がメモリ内ならば、I/Oユニットは読み出し書き込み状態からアイドル状態に移す。しかしながら、アクセスの宛先がI/Oデバイスなら、メモリ制御ユニットは読み出し書き込み状態からアイドル状態に移す。



1. 読み出し/書き込み動作中にプロセッサ・ベースのシステムにおいてクリティカル・パスを削減する方法で、前記システムがメモリ制御ユニット(MCU)、入出力制御ユニット(I/O)、並びに命令集合をそれぞれ備える、前記方法が、

(1) 命令集合から命令の第1の部分をデコードするステップと、

(2) 前記の命令が読み出しと書き込みのうちの少なくとも1つを命じているか否かを決定するステップと、

(3) ステップ(2)で読み出しと書き込みを命じていることが確定すれば、読み出し/書き込み状態に入るために要求をMCUとI/Oの両方に送るステップと、

(4) 読み出し/書き込み動作によってアクセスされるべきアドレスを決定するために命令の残りの部分をデコードするステップと、

(5) アクセス対象のアドレスをデコードするステップで、それは、そのアクセスの対象がメモリかI/Oデバイスかを決定するためにMCUかI/Oの少なくともどちらか一方によって実行されるデコーディングであること、さらに、

(6) アクセス対象のアドレスをデコードするステップで、それが前記の少なくとも2個以上の資源のうちのどれがアクセスされるべきかを決定するために少なくとも1個以上の資源によって実行されるデコーディングであること、さらに、

(7) 前記のデコードするステップ(1)による結果に基づいて要求の残りの部分を取り消すステップと、

から成り立つことを特徴とする方法。

2. 前記ステップ(1)及び(4)が並列で実行されることを特徴とする請求項1記載の方法。

3. 前記のステップ(5)における要求が同時に送られることを特徴とする請求項1記載の方法。

4. 資源へのアクセス動作中にプロセッサ・ベースのシステムにおいてクリティカル・パスを削減するためのシステムで、前記システムが多量の資源並びに1個の命令集合を備え、さらに

(1) 命令集合から命令の第1の部分をデコードするための手段と、

(2) 前記の命令がアクセス動作を命じているか否かを決定するための手段と、

(3) ステップ(2)で資源へのアクセスが求められていることが確定すれば、アクセス状態に入るために少な

(4) 前記のデコードするステップ(1)による結果に基づいてMCUやI/Oに対する前記の要求の一つを取り消すステップと、

から成り立つことを特徴とする方法。

5. 前記ステップ(1)及び(4)が並列で実行されることを特徴とする請求項1記載の方法。

6. 前記ステップ(5)における要求が同時に送られることを特徴とする請求項1記載の方法。

7. 資源へのアクセス動作中にプロセッサ・ベースのシステムにおいてクリティカル・パスを削減する方法で、前記システムが多量の資源並びに1個の命令集合を備え、前記方法が、

(1) 命令集合から命令の第1の部分をデコードするステップと、

(2) 前記命令がアクセス動作を命じているか否かを決定するステップと、

(3) ステップ(2)で資源へのアクセスが求められていることが確定すれば、アクセス状態に入るために少なくとも2個以上の資源に要求を送るステップと、

(4) アクセスされるべきアドレスを決定するために命令の残りの部分をデコードするステップと、

(5) 少なくとも2個の資源に要求を送る手段と、

(6) アクセス対象のアドレスを決定するために命令の残りの部分をデコードする手段と、

(7) アクセス対象のアドレスをデコードする手段で、それはデコーディングが前記の少なくとも2個の資源のうちのどれがアクセスされるべきかを決定するために少なくとも1個以上の資源によって実行されるデコーディングであること、さらに、

(8) 前記のデコードするステップ(1)による結果に基づいて要求の残りの部分を取り消すための手段と、

から成り立つことを特徴とするシステム。

8. 決定の前記手段が、前記命令が読み出し及び書き込み動作の少なくともどちらか1つを命じているかどうかを決定する手段であり、

送るための前記手段が、ステップ(2)において読み出しと書き込みが求められていることが確定すれば、読み出し/書き込み状態に入るためにMCUとI/Oの両方に要求を送る手段であり、

命令の残りの部分をデコードするための前記手段が、読み出しと書き込み動作によってアクセスされるべきアドレスを決定するために、命令の残りの部分をデコードするための手段であり、

アドレスをデコードするための前記手段がアクセス対象のアドレスをデコードするための手段であり、それはアクセスの対象がメモリかI/Oデバイスかを決定するためにMCIUまたはIOUが少なくともどちらか一方によって実行されるデコーディングであること、さらに、

取用した前記手段がアドレスをデコードするための前記手段による結果に基づいてMCIUとIOUへの前記要求の1つを取り消すための手段であることを特徴とする請求項3記載のシステム。

# 発明の名称

メモリ制御ユニット並びに入出力制御ユニットの動作におけるクリティカル・パスの削除  
(Elimination of the Critical Path in Memory Control Unit and Input/Output Control Unit Operations)

## 発明の背景

### 1. 産業上の利用分野

本発明は一般的にはマイクロプロセッサ・ベースのシステムの分野に関し、より具体的には、少なくともユツ以上のサブシステムに読み出し／書き込み(READ/WRITE)要求を同時に送り、引き続いて不要な要求を取り消すためのシステム並びに方法に関する。

### 2. 関連技術

典型的なコンピュータ・ベースのマイクロプロセッサ・システムは、3大サブシステムから成り立っている。それらは、即ち、1個の主記憶、少なくとも1個以上の内部処理ユニット(IPU)、そして1つの入出力(I/O)サブシ

ステムである。従来のマイクロプロセッサ・ベースのシステムの中には、性能を上げるためにパイプライン方式として知られる方法を使うものがある。パイプライン処理では、命令実行の歩みをステップ(例えば、フェッチ、デコード、及び実行)はパイプライン・ステージと呼ばれる独立した単位によって実行される。そのステップは、プロセッサが一度に少なくとも1つ以上の命令を処理できるように、様々なステージで並列で実行される。

パイプラインのスループットは、どの位の程度で1つの命令がパイプラインから出る(exit)かによって決定される。パイプライン・ステージは相互につながっているから、全てのステージが同時に進行するように制御されていなければならない。しかしながら、現実には、フェッチ、デコード、実行のステージの中には他に比べて時間が掛かるものもある。そのために、其の並列性が実行されているとは言えない。1サイクル時間に割り当てられる総時間は最長ステップを実行するのに掛かる時間である。或いは、そのステップを連続のサイクル期間で実行することもできる。いずれの場合でも、システムの効率是最長ステップによって下げられてしまう。

パイプライン方式の詳細については、Hennessy他著の「Computer Architecture: A Quantitative Approach」(Morgan Kaufmann 出版社、1990)を参照のこと。

ソース・プログラムの動作中、IPUは主記憶もしくはI/O空間のデータにアクセスすることが必要になる。例

して、IPUは、プロセッサのサイクル時間が数十ナノ秒の、マイクロプロセッサ・ベースのシステムで最も速い要素である。その一方、主記憶やI/Oデバイスのサイクル時間は数百ナノ秒である。従って、マイクロプロセッサの性能を制限する要因の一つはメモリ・アクセスの速度ということになる。

従来のシステムにおいては、命令デコードの初期の段階から、プロセッサはI/Oまたはメモリへのアクセスが必要かどうかわかっている。しかしながら、命令が読まず最後まで完全にデコードされるまで、アクセス対象が主記憶なのかそれともI/Oなのかシステムは決定できない。従来のシステムでは、読み出し／書き込み状態に遷移するためのコマンドをメモリ制御ユニット(MCU)またはI/O制御ユニット(IOU)に送る前に、OPCODE(演算コード)及びレジスタ・アドレス部分を含む命令体をデコードする。言い換えれば、従来のシステムはこうした処理を逐次に行い、命令の完全なデコードのステップと、MCUまたはIOUを読み出し／書き込み状態に遷移するステップは「クリティカル・パス」上にある。その結果、従来のシステムは、メモリやI/Oデバイスに実際にアクセスするという時間のかかるステップに入る前に、命令を最後までデコードし終るのを待って貴重な時間を浪費しているのである。

典型的なクロッキング動作に對いて、システム・クロックにはスレープ状態として知られる状態がある。

MCUやIOUなどのスレーブ・デバイスが、読み出し／書き込みなどの動作を実行するためにバスへのアクセスが許可されるのは、このスレーブ状態の間だけである。従って、クロックのサイクル時間は、アクセス対象のデバイスが年度のスレーブ・サイクルに間に合うように読み出し／書き込み状態に入れるように、クリティカル・バス上の全ての処理が実行されるのに十分な長さでなければならない。クリティカル・バス上の処理が全てその時間内では実行しきれないようにクロックのサイクル時間が短縮されていれば、読み出し／書き込み動作は年度のスレーブ・サイクルに間に合わず、その次のスレーブ・サイクルまでさらに1クロック期間待たなければならない。従って、必要なのは、アクセス対象の装置のアドレスがデコードされるのを待たずに、OPCODEがデコードされ装置プロセッサが直ちにメモリに対し読み出しまたは書き込みの要求を出せるようにするシステム並びに方法である。こうすることによって、読み出し／書き込み要求をクリティカル・バスからバテレル・バスに移すことによってクリティカル・バスの長さが短縮される。この方法で、システムのクロック・レートが定まり、メモリ及びI/Oサブシステムの実効レイテンシが短縮される。

図1のデコードでメモリがアクセス対象であったことが明らかになると、IOUはアイドル状態に戻り、一方、図1のデコードでI/Oデバイスがアクセス対象であったことが明らかになれば、MCUはアイドル状態に戻る。

## 発明の概要

本発明はプロセッサ・ベースのシステムにおけるクリティカル・バスを短縮するためのシステム並びに方法を提供するものである。本発明に基づく技法においては、命令の演算コードがデコードされ、その演算コードがシステム・メモリまたはI/Oデバイスにアクセスを命じる動作を引き出すと、読み出し／書き込み要求が直ちにプロセッサ・ベースのシステムのメモリ制御ユニットMCUと入出力制御ユニットIOUの両方に送られる。プロセッサが命令の残りの部分をデコードする前にこの読み出し／書き込み要求は送られるため、現在のクロックが終る前にMCUとIOUがアイドル状態から読み出し／書き込み状態に遷移することが可能になる。

この技法によって、サイクル時間は、プロセッサが現在の命令をデコードするのにかかる合計時間のみに抑えられるようになる。さらに時間を追加することなく現在のクロック・サイクル時間内で、MCU並びにIOUは読み出し／書き込み状態に遷移できる。それは何故ならば、MCUとIOUは既にその状態に入っているからである。

命令が最後まで残らずデコードされ、メモリ・アクセスまたはI/Oアクセス用のアドレスが計算され、さらにMCUまたはIOUの宛先がデコードされたら、取消し処理が始まる。取消し処理では、MCUまたはIOUのいずれかが読み出し／書き込み状態からアイドル状態に戻される。

## 図面の簡単な説明

本発明については従の背景請求の範囲で具体的に説明されている。今までに述べたさらにそれ以外の本発明の優れた点については、添付の図面と共に以下に記述する説明を参照することによってより理解が深まるであろう。

図1図は本発明に基づくシステム・アーキテクチャI/Oの概略ブロック図である。

図2図はシステムのメモリ・マップである。

図3A図は典型的な命令フォーマットを説明したものである。

図3B図は、2個のレジスタに対する1つの演算の結果を基に1つのアドレスが計算される典型的な命令フォーマットを説明したものである。

図4図はクリティカル・バス上の全ての項目を含む従来のシステムの状態動作の流れを示す図である。

図5図は本発明の技法に基づく例を示す状態図である。

図6図は本発明によるIOUの取消し手続のためのタイミング図である。

図7図は本発明の技法を示す状態図である。

# 図1

第1図は本発明が適用できるアーキテクチャを代表する、マイクロプロセッサのアーキテクチャ100の概略図を示したものである。第1図において、本発明の好適な実施例に基づいて、100と指定されたマイクロプロセッサ・システムのアーキテクチャが提示されている。システム・アーキテクチャ100は、内蔵プロセッサ・ユニット(IPU)105、メモリ制御ユニット(MCU)120、I/O制御ユニット(IOU)130、主記憶150、及び外部I/Oデバイス160から成る。

主記憶150は、好適な実施例においては、32ビット幅のメモリ・バス140によってMCU120に接続されている。尚ほ、I/Oデバイス160は、好適な実施例においては、32ビット幅の外部I/Oバス135によってIOU130に接続されている。

MCU120はIPU105と主記憶150間の読み出し書き込み動作を制御する回路である。本発明に基づく好適な実施例におけるMCU120は、スイッチ・ネットワーク121及びポート122とから成っている。本明細書に記載の図においてはポート122は1個だけしか示されていないが、本発明に基づく好適な実施例は複数のポート122を備えている。

また、IPU105とポート122の要求を調整する手段としても設けられる。ポート122(調整または調整)は、I/Oデバイス160にデータを送り、且つそこからデータを受け取る役目を果たす。

第2図はシステム・メモリの構成を示したものである。好適な実施例において、メモリ構成は2領域に分割されている。I/Oバス領域210とメモリ・バス領域220がある。この構成の結果、システムは主記憶150とI/Oデバイス160の両方に対する読み出し書き込み動作を要求することができるようになる。読み出しまたは書き込みの動作を命ずる命令は、システム・メモリ・マップ200上に与え可能なアドレスを含むことになる。

本発明に基づく好適な実施例において、命令処理は3ステージのパイプラインに分割されている。その3ステージとは、フェッチ、デコード、並びに実行である。最初、命令はIPU105によってメモリからフェッチされる。その後、命令は指示された機能(例えば、加算、減算、格納)を規定するためにデコードされる。3番目且つ最後、そのデコードされた命令はその機能の通りに実行される。

命令をコード化している時、レジスタ及びアドレス指定モードは共に命令サイズに大きな影響を及ぼす。これはアドレス指定モードのフィールドとレジスタ・フィールドが1つの命令の中で所収も現われることがあるからである。事実、ほとんどの命令が、演算コードの指定よ

リスイッチ・ネットワーク121にMCU120とIPU105間の通信インタフェースを可能にする。スイッチ・ネットワーク121はまた、ポート122の要求を調整する手段としても設けられる。ポート122(調整または調整)は、主記憶150にデータを送り、且つそこからデータを受け取る役目を果たす。

スイッチ・ネットワーク121の機能は、IPU105及びIOU130からの様々な命令並びにデータ要求を受け取ることである。IPU105及びIOU130はメモリ要求者と称されることがある。要求を受け取り調整した後に、要求は対応するメモリ・ポート122に送られる。対応するポート122の決定は命令アドレスの解釈によってなされる。ポート122(調整または調整)は次に必要なタイミング信号を生成し、外部メモリ150のアドレス指定された部分にデータを送るか、またはそこからデータを受け取るかとする。

IOU130はIPU105とI/Oデバイス160間の読み出し書き込み動作を制御する回路である。本発明に基づく好適な実施例におけるIOU130は、スイッチ・ネットワーク131及びポート132とから成っている。本明細書に記載の図においてはポート132は1個だけしか示されていないが、本発明に基づく好適な実施例は複数のポート132を備えている。MPUの場合と同様に、IOU130のスイッチ・ネットワーク131、IOU130とIPU105間の通信インタフェースを可能にする。スイッチ・ネットワーク131

りもアドレス指定モードやレジスタ・フィールドのコード化に密着にビットをとられている。このように、アドレス指定モードとレジスタ・フィールドが命令ビットの大きな割合を占めるため、それらのコード化は非常に簡単に命令のデコードを実現できるかに大きく影響している。

命令は幾つかのセクションに分割されることが多い。第3A図は命令の一部とその構成要素を示すものである。第3A図において、命令300はOPCODE310を含んでいる。OPCODE310は、一旦デコードされれば、どのオペレーションを実行すべきかプロセッサに伝える。オペレーションにはADD(加算)、COMPARE(比較)、CLEAR(クリア)などの算術演算、BRANCH(分岐)やJUMP(飛び越し)などの制御動作、RETURN FROM INTERRUPT(割り込みからの戻り)などのシステム動作、並びにMOVE(移動)やPUSH(プッシュ)などのデータ転送動作などがある。

レジスタやメモリはI/Oデバイスのどれだけのアドレスは、必要に応じて、プロセッサにどの記憶場所に対してオペレーションを行なうべきか伝えるために、命令の中に包含されている。これらは記憶場所320、330で命令にコード化される。例えば、OPCODE310がADDの演算を行なうようにプロセッサに命令すれば、プロセッサは、加算対象の値を含むアドレスを決定するために、記憶場所320と330を見に行きそこに入っている値を

アコードする（直接アドレス指定の場合）。

命令300の実際のフォーマットはシステム・アーキテクチャによって異なる。どうすれば本説明に基づく技法が他の命令フォーマットに適用できるかは当業者には明らかになるであろう。

第3B図はもう少し複雑なアーキテクチャ用の命令フォーマットを示したものである。第3B図におけるOPCODE310は、第3A図に示したフォーマットにおいて説明したのと同じオペレーションを行なうように、プロセッサに命令することができる。しかしながら、ここでは、アドレスの割当て異なる。このフォーマットでは、最初のアドレスは2つのアドレスをデコードし演算することによって計算される。最初のアドレスを計算するために、プロセッサはRindex340とRbase350をデコードする。実際のアドレスを決定するためにこれらの記憶場所における値に対し演算が行なわれる（この場合、合計される）。このタイプのアーキテクチャは多数のプログラミングに有効である。2番目のアドレスを計算する必要がある場合、記憶場所Raddress360における値がデコードされるだけである。

バイブラインの命令アコード部分は幾つかの段階に分解される。命令のそれぞれ異なる部分、即ち、OPCODE310、Rindex340、Rbase350、Raddress360は様々な段階でデコードすることができる。従って、プロセッサは、命令の残りの部分に先行してOPCODE310を

先にデコードすることができる。一旦命令のOPCODE310部分がデコードされれば、プロセッサは命令がどんな動作を命じてくるかわかる。

従って、例えば、命令は計算または比較であるとデコードされたOPCODE310がIPU103に伝えられ、プロセッサは演算対象の値に値の記憶場所からアクセスしなければならないことを知ると、第1図と第3B図において、従来のシステム用の典型的なシナリオについて説明している。命令はフェッチされ、OPCODE310がデコードされる。OPCODE310がプロセッサにロード／読み出し／書き込みを実行するように命令すれば、IPU103は格納対象の値のある記憶場所を特定するためにRaddress360をデコードする。IPU103は次にRindex340とRbase350をデコードする。一旦デコードされれば、Rindex340とRbase350はデータの処理アドレスを計算するために加算器に送られる。その次に、この最終アドレスはデコードされるためMCU120またはIOU130に送られる。デコードされたら、MCU120またはIOU130のいずれか読み出し／書き込みを実行すべき方が、読み出し／書き込み状態に移される。しかしながら、こうした処理を全て逐次に実行し、しかも今度のスレープ・クロック・フェーズに間に合うようにそれに対応するユニットを読み出し書き込み状態に移させるには、現在のクロック・フェーズ内では時間的に十分とはいかない場合がある。スレープ・クロック・フェーズはその期間中読み出し書き込

み動作が発生しうる唯一のフェーズであるから、最初のスレープ・フェーズに間に合えば、そのシステムは次のスレープ・フェーズまでクロック・同期する必要がある。従って、これらの処理は全て逐次に行われ、それら高度クロック・レートでは直列的読み書きで時間はより次に戻るとスレープ・フェーズに間に合わない原因となるから、クリティカル・パスが存在しているということになる。或いは、パス上の全てのフェーズが完了するように、システム・クロックのサイクル時間を増やさなければならない。しかしながら、この代替案は、処理を遅くすることになるから、望ましい問題解決法とは言えない。

第4図は従来のシステムに関する上記のロード／格納のシナリオに基づくタイミング図並びに状態図を示すものである。下記の動作時間は例として記述するものである。勿論、他のマシンでは命令処理との関連で演算時間は異なってくる。第4図において、第1クロック期間T0/T1で、ロード／格納命令がフェッチされる。第2クロック期間T2/T3で、命令がデコードされる。即ちOPCODE310がデコードされ、その値Raddress360、Rindex340、さらにRbase350がデコードされる。Rindex340及びRbase350はT3のクロック・フェーズでデコードされる。この処理はおよそ8ナノ秒かかり、ブロック410で示されている。この処理が完了すると、アドレス計算が行われる。この計算に要する時間はブロック

420に示すように7ナノ秒である。アドレス計算実行後に、アドレスはMCU120もしくはIOU130によってデコードされなければならない。ブロック430で示した、このデコーディングは4から5ナノ秒かかる。従って、命令のデコーディングにかかる時間は合計で20ナノ秒である。

アドレスがブロック430でデコードされた後、MCU120またはIOU130のいずれか対応する制御ユニットは、次のスレープ状態より読み出し／書き込み状態に入っていなければならない。第4図において、T4、T6、T8などはスレープ状態、即ち、メモリI/Oの読み出し／書き込みが起きる唯一の状態である。命令とアドレスのデコーディング動作がその前のT3でクロック時間の20ナノ秒を消費し、且つ我々は今度のスレープ・フェーズT4で読み出し／書き込みが起きるようにしたいため、クロック・フェーズの期間は20ナノ秒よりも長くしなければならない。20ナノ秒を超えてさらに要する時間はMCU120またはIOU130の対応する制御ユニットが読み出し／書き込み状態に移るのににかかる時間である。そのフェーズの時間が足りなければ、MCU120またはIOU130はT4のクロック・フェーズに間に合うように読み出し／書き込み状態に入ることができず、クリティカル・パスが存在し、さらにシステムは読み出し／書き込み動作を実行するためにT6のクロック・フェーズになるまで待たなければならない。しかるに、従来のシステムの欠陥は、命令アコード動作及びメモリまたはI/Oバイ

スの読み出し／書き込み状態への移行が正常に起こり、且つ高速クロック・レートでクリティカル・バスを動作することである。その結果、クロック・サイクル時間の増大（クロック・レートの低下）を招いたり、その間読み出し／書き込み動作を延期しなければならぬ「時間の無駄使い」のクロック期間に陥ることになる。この後者の結果は、後者はさらにクロック・サイクル短くないと、即ち次のパイプラインまで、実行完了しないということである。

## 説明

本発明では、アドレス・デコード動作に先行してMCU120とIOU130を読み出し／書き込み状態に遷移することによって、このクリティカル・バスの欠陥を回避する。言い換えれば、アイドル状態から読み出し／書き込み状態への移行のステップがクリティカル・バスから取り除かれるということである。第3図は本発明に基づく動作を固めたタイミング及び状態の図である。第10及び第5図において、デコードされたOPCODE310がメモリまたはI/Oデバイスに対する読み出し／書き込み動作が起まなければならないことを示していれば、IOU130の読み出し／書き込み要求320はOPCODE310のデコード直ちにMCU120並びにIOU130に並列で送られる。IOU130は他のレジスタがデコードされるまで待つこと

はしない。この方法では、命令の残りの部分がデコードされている間に、MCU120及びIOU130はT3のクロック時間中にアイドル状態540から脱し、読み出し／書き込み状態に入る。従って、読み出し／書き込み状態に遷移する動作は遅延ではなくなるから、システムはT4のフェーズ発生まで待たされることがなくなる。即ち、クリティカル・バスがなくなる。今や、MCU120及びIOU130は、スレーブのクロック・フェーズT4発生より前に読み出し／書き込み状態330に入っている。

この概念は、従来のシステムについて先に説明したのと同じタイミングの数字を使って表すことができる。例えば、レジスタのデコーディング、アドレス計算さらにアドレスのデコーディングに要する時間が70ナノ秒以下ならば、遅延20ナノ秒のクロックを流すシステムではクリティカル・バスは存在しない。それは何故ならば、IOU130及びMCU120はT4の発生以前に既に読み出し／書き込み状態に入っているからである。

最先がデコードされ、MCU120とIOU130のどちらが必要なのかが分かれば、IOUの取扱い番号560が生成され、従って不要の追加ユニットをアイドル状態に戻す。

同じ第3図並びに第5図において、OPCODE310のデコードは、ブロック310で示すように、T2、T3のクロック期間の初期に発生する。このデコードが完了し次第直ちに、IOUの読み出し／書き込み要求320がMCU120とIOU130に送られる。要求320はMCU120を読み出し／

書き込み状態330Aに、IOU130を読み出し／書き込み状態330Bにする。読み出し／書き込み状態330は共にT4のスレーブ・クロック・フェーズ中の低レベルに合うように既に発生している。

読み出し／書き込み要求と同時に、命令の残りの部分が、対応するアドレスを確定するためにデコードされる。Rindex340並びにRbase350のデコーディングはブロック312に示す通りである。アクセス対象のアドレスの計算。即ちRindex340のRbase350への加算はブロック314で示す通りである。

制御ユニットMCU120またはIOU130の一方が読み出し／書き込み動作を実行し、他方はしない。ブロック350に示すように、アドレス・デコードが起きるまでどちらがどちらなのか不明である。アドレス・デコードが起きるとすれば、IOUの取扱い番号560が生成され、従って読み出し／書き込み動作を行っていないユニットを取り消す。このユニットはアイドル状態540に戻る。

第5図に示した図において、アクセス対象のアドレスは、たまたまメモリ内に存在している。従って、そのアドレスがデコードされると、I/Oデバイスはアクセスされるべきでないことが分かる。次に取扱い番号560が生成され、IOU130はアイドル状態540Bに戻る。MCU120は読み出し／書き込み状態330Aのままで、メモリ・アクセスが発生する。

非過な負荷状態において、MCU120はIOU130によっ

て計算されたアドレスをデコードする。アドレスがデコードされ、アドレスがメモリ150とI/O160のどちらに宛てられているかが判明すれば、MCU120は自分の読み出し／書き込み要求を取り消しアイドル状態340Aに戻るか、もしくは取扱い番号をIOU130に送るかのいずれかを行う。このアドレス・デコードはMCU120またはIOU130のいずれか、或いはその両方によって実行されることがあるということは当業者には明瞭である。

本発明に基づく往時のタイミング図は第6図に示す通りである。第6図において、IOU\_REQUEST番号320はMCU120及びIOU130の両方に送られる。MCU120は許可番号610Aに回答し、IOU130は許可番号610Bに回答する。MCU120とIOU130は共に、MCU\_STATE620A及びIOU\_STATE620Bに示すように、読み出し／書き込み状態にある。一旦アドレスがデコードされれば、MCU120またはIOU130のいずれか不要のユニットを取り消すために、MCU/IOU\_CANCEL番号が生成される。第6図はIOU130が不要で、従ってIOU\_STATE620Bがアイドル状態に戻るケースを示したものである。

第7図はMCU120とIOU130の状態を示すものである。第7及び第8図において、取扱い論理ブロックを生成するために使われる状態マシンが示されている。取扱い論理はMCU120にもIOU130にもある。T<sub>1</sub>でIOU130からの読み出し／書き込み要求番号を受信すると、その1つの状態マシンは同時に動作を開始する。次の序元（即ち、



現在デコード進行中の命令に示されるメモリ内のアドレス}のデコードは、その動作を取り消すために、MCU120またはIOU130のいずれかが、T<sub>T</sub>でスイッチ125もしくはスイッチ133からそれぞれ取崩し信号を生成する。要求が取り消されれば、取り消された制御ユニットはアイドル状態に戻る。要求が取り消されなければ、制御ユニットは対応するメモリの記憶場所にアクセスし始める。

取崩し信号は命令の第21番目のビットを検査することによってトリガされる。そのビットがゼロであれば要求は主記憶域に対するもので、そのビットが1ならその要求はI/Oデバイスに対するものである。第2図に示すように、アドレスが0000\_0000と03FF\_FFFFの間であれば、そのアドレスはメモリにアクセスする。しかしながら、アドレスがFF80\_FFFFとFFFF\_FFFFの間であれば、I/Oにアクセスする。好適な実施例において、他の領域は全て予約されている。本発明において他に多数のメモリ・マップが採用できるということが当業者には明確であろう。

再び第2図において、好適な実施例で、MCU120がアドレスをデコードすると、そのアドレスが予約領域230の1ヶ所に宛てられているかどうかも特定される。予約領域230の1ヶ所に宛てられていれば、IOU130及びMCU120に対する読み出し/書き込みは共に取り消される。

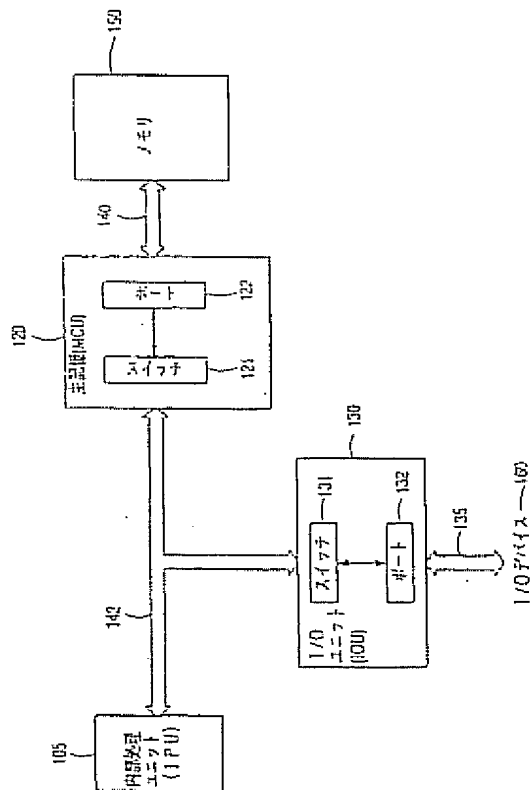
## 効果

従って、本発明はシステムのクロック・レートを選めることができる技法を提供するものである。OPCODEがデコードされ、読み出しもしくは書き込みの動作が呼び出されことを示していれば、直ちに読み出し/書き込み状態に過渡するようにメモリ及びI/Oユニットに命令を送ることによってクロック・レートは決められ、かくしてクリティカル・パスから状態遷移時間が取り消される。

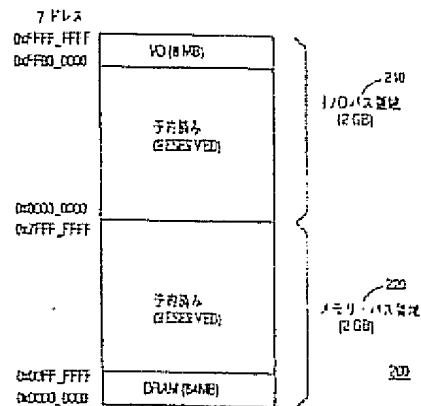
本発明について好適な実施例を挙げて具体的に説明してきたが、本発明の範囲並びに適用範囲から外れることなく形状及び詳細において様々な変更が可能であることが当業者には理解されることであろう。

例を挙げれば、本発明に基づく技法は本明細書に記載のMCUやIOU以外の資源デバイスを使ってシステムで実行することが可能であり、且つ本発明による技法は多数の資源デバイスを包含するシステムに拡張することができる。

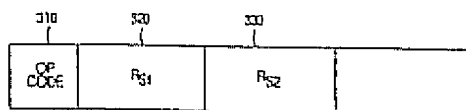
この発明を実行する際には、トレードオフがある。MCU120及びIOU130の片方が読み出し/書き込みモードにならざるに要求される時でも、MCU120及びIOU130は共に、読み出し/書き込みモードになる消費電力が増加する。高速クロック・レートにすることによってシステムの性能を上げれば、ほとんどのアプリケーションでこの消費電力の増加は相殺される。



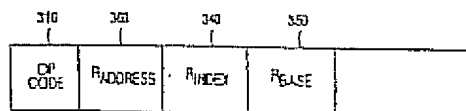
第1図



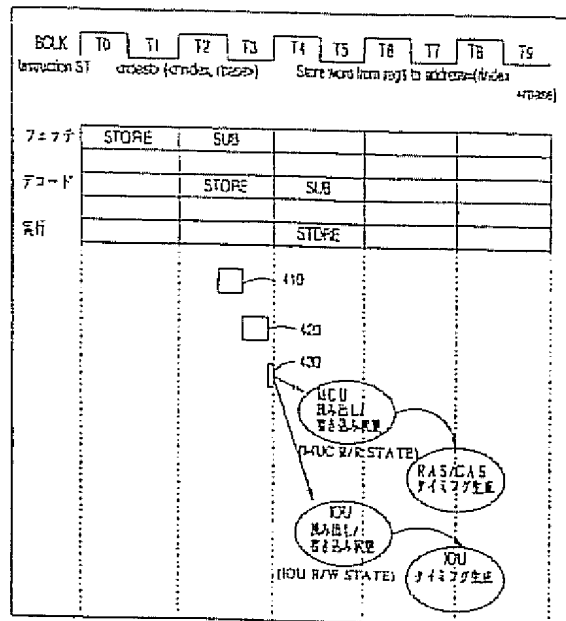
第2図



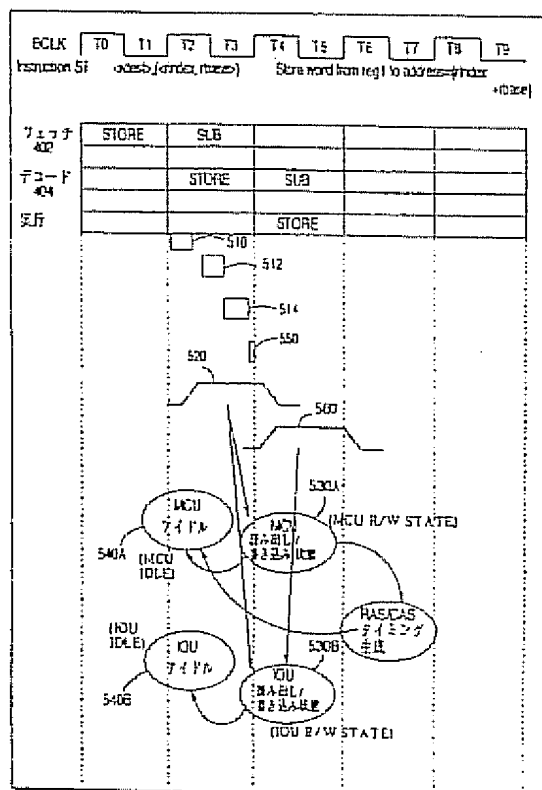
第3A図



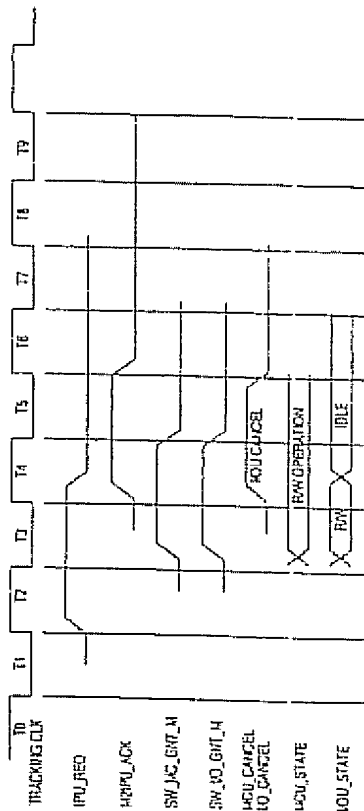
第3B図



第4図



第5図



第6図

